QCVis: A Quantum Circuit Visualization and Education Platform for Novices

QCVis:
A Quantum Circuit Visualization and Education Platform for Novices

a thesis presented
by
Milan Marie Williams
to
The Departments of Computer Science and Physics

in partial fulfillment of the requirements
for the degree of
Bachelor of Arts
in the subjects of
Computer Science and Physics

Harvard University
Cambridge, Massachusetts
May 2021

Thesis advisor: Dr. Johanna Beyer Milan Marie Williams

QCVis:
A Quantum Circuit Visualization and Education Platform for Novices

Abstract

QCVis is a quantum circuit visualization and education platform for novices. The rapid growth of the quantum computing field surpasses the educational capacity of existing systems, which makes novices particularly in need of accessible tools. Often, novices struggle to understand two fundamental concepts: how quantum gates affect both qubit and quantum circuit state probabilities over time. QCVis contributes two novel visualizations to develop this intuition. One visualization teaches users about the relationship between gates and single qubit probabilities. This feature integrates stacked bar charts into the traditional quantum circuit diagram to display the state probability of a single qubit after each gate application. The other visualization provides insight into the intermediate steps that contribute to the final state probability distribution. The combination of a radial bar chart and a time slider helps users see the state probability of a quantum circuit at a user-specified execution step and observe how it changes over time. We evaluated the design of QCVis in a user study with novices, who successfully used our interface to understand these key quantum computing concepts.

Contents

Listing of figures

This thesis is dedicated to my Mom and Dad, my first science teachers.

Acknowledgments

This thesis is the culmination of my four years at Harvard and the result of over a year of research. There are countless wonderful people who encouraged me along the way and I am grateful for your support.

I would like to express my deepest appreciation to Dr. Johanna Beyer for her support as my thesis advisor and personal mentor. I am grateful for her genius insights, kind words, and endless support throughout my journey as a budding researcher. Without her, this thesis would be impossible.

I would like to thank Professor Prineha Narang for her support as an educator and personal advisor. I am grateful for her quantum computing brilliance, abundant generosity, and limitless chocolate treats.

I would like to thank Elisa Zhao Hang, Adinawa Adjagbodjou, and Robert Krueger whose crucial contributions to this project propelled it forward.

I would like to thank Mom and Dad for their unwavering support over the past four years. We did it!

Finally, I would like to thank countless classmates, whose friendships and contributions made it all worthwhile.

0

Introduction

0.1 Overview

THE study of physics is a quest for answers about nature, about its past, present, and future.[26] The study of computer science is a quest to understand and conduct computational analysis. Quantum computing, the brain-child of these academic fields, makes use of nature's quantum mechanical laws in order to conduct advanced computational analysis. Quantum computers differ from the classical computers we use daily like smartphones, tablets, or laptops. For years, theorists claimed that quantum computers could solve problems that are not practically possible for a classical computer to solve. Google and other research teams recently proved this theory, known as quantum sumpremacy, in late 2020[2]. This demonstrated to the world that quantum computers may become the most powerful form of computation yet. Quantum computers conduct this high powered computational analysis on qubits (quantum bits), the basic unit of information in a quantum computer. Due to this high power, quantum computing is expected to grow quickly and advance many fields of study. Estimates say it will reach a global market value of 1.7 billion by 2026, up from a 472 million market value in 2021.[21] Initial applications appear to be in banking and finance[21], with more to come in artificial intelligence[24], molecular modeling[7], cryptography[14], to name a few.

0.2 Motivation

With fast paced growth, comes growing pains. The need for many trained quantum computer scientists comes from the industry's rapid rate of expansion. Already, CEOs of quantum computing companies have noted a lack of trained quantum computer scientists in hiring[19]. However, training large volumes of quantum computer scientists at a fast rate is no easy task. The most common quantum computer scientist training pathway is through advanced educational programs like undergraduate and graduate degrees. These paths are robust and reliable, yet require years of training and are accessible to a much smaller audience, given that only 18% of Americans have a STEM degree.[27] More so, the general public is unable to grasp quantum computing's most basic concepts. For instance, fewer than half of Americans know that an electron is smaller than an atom[6]. Tools that aim to teach quantum computing must assume little to no prior familiarity with the field and its related subjects (math, physics, etc.) Therefore, we have established grounds for a new way to educate novices about quantum computing concepts, motivated by the fast growth of the field and lack of accessible educational pathways.

However, in the multitude of quantum computing concepts, which is the most central for these educational purposes? While there is no clear consensus, we propose that quantum circuits are a strong contender. Quantum circuits are the code of quantum computers. They act on qubits to perform and output quantum computation. The building blocks of quantum circuits are quantum gates, a fundamental quantum operation (ex: rotation) that applies to a small number of qubits. Quantum gates applied in succession make up a quantum circuit. Quantum circuits output a single state (0 or 1) for each qubit in the circuit. However, the same quantum circuit may not output the same result every time it is run. To predict what the quantum circuit will output is to understand the quantum circuit's state probability. The state probability of a quantum circuit describes the state probability of each qubit. It describes how likely a qubit is to return a 0 or 1. Additionally, the state probability of a quantum circuit is not constant throughout the computation time. It changes overtime as new gates are applied. In short, quantum gates change an individual qubit's state probability, which, in turn, changes the quantum circuit's state probability. Detailed explanations of these and other quantum concepts can be found in Chapter 1. Through this discussion, we have identified two central tasks that a newcomer must complete in order to understand quantum circuits, and by relation, quantum computing.

> **T1**: A user must learn how quantum gates change the state probability of a single qubit over time.
>
> **T2**: A user must learn how quantum gates change the state probability of the overall quantum circuit over time.

Understanding these two concepts will help users create intuitions and gain knowledge about quantum computers.

0.3 Related Work

In order to meet the intellectual demands of the field, nontraditional educational methods are necessary to quickly onboard newcomers. For this thesis, we focused on creating novel visualizations to better explain the complexities of quantum computing to novices. Visualizations can "refer to the process of creating visual imagery" or "the process of interpreting in visual terms or of putting into visual form".[30] Put another way, visualizations give new meaning to information as they encourage the formation of new mental images. How visualizations do this is a fascinating field of research, though out of scope for this thesis. For those interested, Winn[32] gives a wonderful survey into the psychology behind visualizations and education. Regardless of the specifics, it is intuitively understood that visualizations are an excellent learning tool, especially in science. The science education community has always understood that visualizations impact educational development. It is generally agreed upon that "visualization objects assist in explaining, developing, and learning concepts in the field of science"[18]. Visualizations appear to help people learn and are a reliable method to teach people about scientific concepts. With this in mind, we define the key qualities of a scientific visualization. Based on 20 years of research in science education[30 11], a successful science education visualization has two goals:

> **G1**: A successful visualization will promote learning, understanding, analysis and problem solving.
>
> **G2**: A successful visualization will be tested by students who will validate its usefulness.

These goals motivate the entirety of this thesis.

Since the inception of quantum computing, formal educational institutions have adopted a variety of visualizations to teach quantum fundamentals. The first and most common is the Bloch sphere[5] a standard 3D model for visualizing the state of a single qubit (Figure 1).

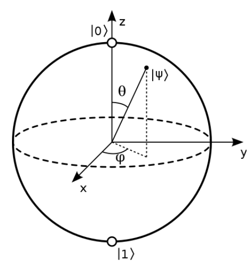


Figure 1: The Bloch sphere and a state $|\varphi\rangle$.

In theory, a qubit can hold an infinite number of states. All of these states are uniquely represented as a vector on the Bloch sphere, as a sphere has an infinite number of points on its surface. However, the third dimension of the Bloch sphere introduces the possibility of visual occlusion. As defined by Szafir in *Five Ways Visualization Can Mislead*, "visual occlusion occurs when some marks [in the visualization] make it difficult, or impossible to view others"[28], a phenomena that commonly occurs in 3D visualizations. In the Bloch sphere, both a vector pointing into the page and a vector pointing out of the page, appear to represent the same state. This demonstrates the limitations of the Bloch sphere as a scientific visualization for quantum computation education.

Another educational visualization is the quantum circuit diagram[3] that models gate-based quantum computation (Figure 2).

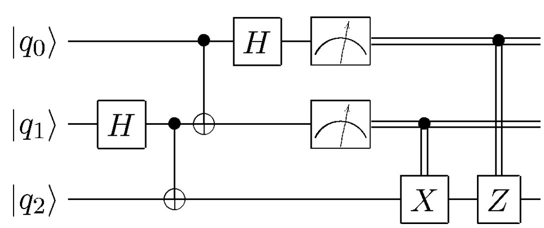


Figure 2: An example quantum circuit diagram.

We'll provide an in-depth explanation of the quantum circuit diagram in Chapter 1. Just like the Bloch sphere, they are used in textbooks and are praised for their visual simplicity. While they are widespread, their minimalist design means that they are not visually intuitive to novices without additional context. In our preliminary user testing, we found that they lack the informational context and interactivity to enable users to do meaningful scientific learning and independent problem solving.

While historically the largest, educational institutions are not the only player in the quantum computing visualization space. Major technology companies, including IBM [8], have also delved into quantum computing visualizations, and more specifically, quantum circuit composition interfaces. However, in our survey of existing platforms, they mainly provide a basic interface for users to access the company's quantum computer hardware. They focus very little on educating novices about the conceptual frameworks behind quantum computing.

Overall, current visualizations struggle to intuitively and concisely display quantum computation, especially through quantum circuits, our central model for quantum computation. There is room to develop a new educational tool to help people learn about quantum computers through quantum circuits.

0.4 Contribution

The purpose of this thesis is to present *QCVis*, a novel quantum circuit visualization and educational platform designed to teach novices about quantum circuits. The contributions of this thesis are based on the important attributes of a quantum circuit and educational visualization:

1. We contribute a novel visualization of a single qubit's state probability over time through stacked bar charts that display the state probability in the quantum circuit.

2. We contribute a novel visualization of the quantum circuit state probability over time through a dynamic radial bar chart.

3. We contribute a user study with novices that demonstrates QCVis' higher educational outcomes over a comparable interface.

The thesis will be structured as follows: Chapter 1 provides a detailed background of quantum computing and its key concepts. Chapter 2 conducts a task analysis for the platform, detailing the high-level goals and specific tasks of the project. Chapter 3 presents the original design and implementation of QCVis. Chapter 4 discusses the user study procedure and results. Chapter 5 presents the final design of QCVis. Chapter 6 concludes and summarizes all of the above.

1

Quantum Computing Background

1.1 Introduction

This thesis aims to create scientific visualizations for the purpose of quantum computing education. A solid understanding of the central concepts in quantum computing is essential to understand the contributions of this thesis. Therefore, we dedicate a significant portion of the thesis to a thorough background of key quantum computing concepts. As established, quantum computing uses quantum mechanics in order to conduct high-power computations. More specifically, quantum computing uses two main quantum mechanical principles to power its computation. These quantum mechanical principles are *superposition* and *entanglement*. Both of these concepts are described in detail in the following sections. With the added understanding of superposition and entanglement, we transition into a deeper discussion about quantum circuits. Our quantum circuit discussion is divided into the four components of a circuit: input, gates, output, and error correction.

1.2 Superposition

The first physical phenomena that powers quantum computation is called superposition. Put simply, an object in superposition exists in different states at the same moment in time. A common explanation of superposition is through the Schrödinger's Cat thought experiment[29]. It is described as follows:

> Picture a closed box. Inside the box is a cat, a detector, a radioactive source (say, a decaying atom), and a bottle of poison. If the detector is activated, the bottle of poison is broken, and the cat dies. The detector is activated by the radioactive source, an atom's decay. We assume the radioactive source is small enough such that, over the course of an hour, the atom will decay 50% of the time. After an hour, is the cat dead or alive?

After puzzling for a minute, most people realize there is not a clear answer. There is a 50% chance that the cat is alive and a 50% chance that the cat is dead. Schrödinger says that the cat is in a superposition. The cat is both dead and alive. The cat exists in two different states at the same moment in time.

What happens when you open the box? Well, the observer determines the answer to the question: the cat quickly leaps out of the box, or it lies dead. The state of the cat is clear. In quantum mechanics, to "open the box" is to collapse (measure) the state. It means to determine with 100% probability the state of the system. Once an object's state is collapsed, the superposition is destroyed.

In quantum computers, the objects in superposition are qubits (quantum bits). Qubits are the "fundamental unit of quantum information"[25]. Quantum computers store and compute information in qubits. Qubits are the quantum parallel of the 0 and 1-based bits found in classical computers. Formally, a qubit is a two-state system, where the two possible states are 0 and 1. These states are in Dirac (bra-ket) notation are:

$$|0\rangle \text{ and } |1\rangle$$

Dirac notation is the most common notation for quantum states and is used frequently throughout quantum mechanics[9].

Mathematically, the state of a qubit $|\varphi\rangle$ is:

$$|\varphi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where

$$|\alpha|^2 + |\beta|^2 = 1$$

Theoretically, a qubit can store an infinite number of states, as there are an infinite number of combinations of α and β that satisfy the above equation. This partly explains how a quantum computer can store much more information and compute much faster than classical computers. A physical example of a qubit is an electron, where its spin-up and spin-down states satisfy the two state requirement for a qubit[12].

1.3 Entanglement

The second physical phenomena is entanglement. Quantum entanglement is when two or more objects interact such that the state of one object is intrinsically connected to the state of the other object. To understand this more concretely, we modify Schrödinger's thought experiment.

> Picture two closed boxes. Inside each box is a cat and a bottle of poison. In between the boxes, there is a detector and a radioactive source. As before, we assume the radioactive source will decay 50% of the time. As a result, there is a 50% chance that the detector is activated by the radioactive source. If the detector is activated, then it releases the poison in Box 1 and ignores the poison in Box 2. Cat 1 dies and Cat 2 lives. Alternatively, if after an hour the detector is not activated, it ignores the poison in Box 1 and breaks the poison in Box 2. Cat 2 dies and Cat 1 lives. After an hour, which cat is dead?

As before, there is no clear answer and even worse, one cat is guaranteed to die. In this scenario, the observer cannot know exactly which cat died without opening the boxes. Before the box is opened, both cats are in an equal superposition of the dead-alive states.

However, after opening only Box 1, the observer can determine the state of both cats. If a cat leaps out of Box 1, then the observer knows that Cat 2 is dead. Similarly, if the observer opened Box 2 and saw a dead cat, then a live cat is certain to be in Box 1. If the state of one cat is known, then the state of the other cat is known as well, even without opening its box. By the set-up of the thought experiment, the two cats are inextricably linked - an entangled pair.

Quantum computers entangle and collapse qubits to perform computations. In an entangled qubit pair, once one qubit's state is collapsed, the other qubit's state is also collapsed. After measurement, both the superposition and entanglement are destroyed.

With a deeper understanding of the quantum mechanics behind quantum computers, we can now define a few additional key components of quantum computation.

1.4 Quantum Circuits

Now that we understand *what* quantum computers are, we can learn *how* to use them. Parallel to their classical analog, we can write programs in order to use quantum computers. More specifically, we can write programs which act on qubits and perform quantum computation. The most common model for these programs is called a quantum circuit[3]. Its name is inspired by the classical circuit[31] in computer science. Figure 1.1 is an example quantum circuit.

For the purposes of this thesis, we will treat the quantum circuit model as our main representation of quantum computation. However, it is worth noting that it is an abstraction of the physical system that implements the quantum gates. There is remaining work to develop educational material for lower level quantum computation[1].

Formally, a quantum circuit q contains a number of qubits n. The circuit q has n inputs and n outputs, where n can be any whole number greater than 1.

A quantum circuit can be broken down into four key components.

1. Input
2. Gates
3. Measurement (Output)
4. Error Correction (Optional)

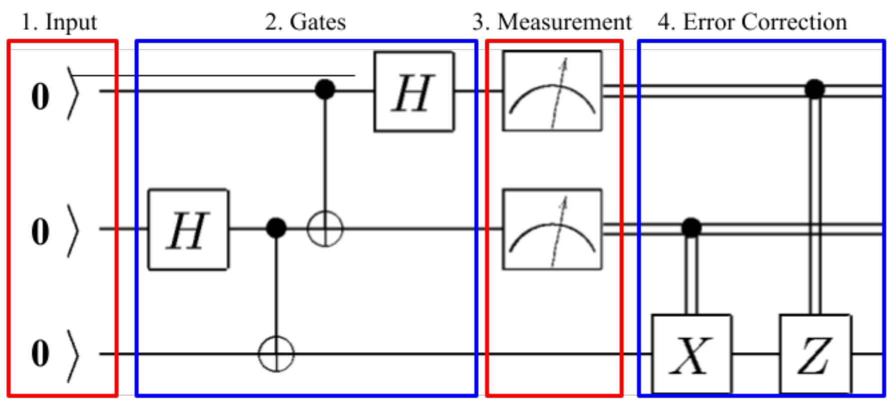


Figure 1.1: An example quantum circuit diagram segmented by its four components: input, gates, measurement, and error correction.

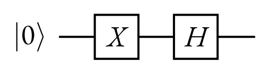
1.4.1 Input

By definition, quantum circuits have n input qubits. In a quantum circuit diagram, these n qubits are represented by a column of $|0\rangle$ kets. The length of the column is always n. This represents how all qubits are initially set to $|0\rangle$ (the 0 state) at the start of every quantum computation. If no further gates are applied and all qubits are measured, then all outputs will be 0.

1.4.2 Gates

A quantum gate is represented by a letter inside a square on this circuit diagram (Figure 1.1), though the exact gate symbol differs depending on the type of gate. A list of all common gate symbols is listed in the Appendix. A quantum gate is the most basic building block of a quantum circuit. It is a fundamentally physical operation that acts on one or more qubits and is time reversible. Most gates affect 1 to 3 qubits.

In this quantum circuit diagram, there are two gates:



The first gate is a Pauli-X gate [17], the quantum version of the classical NOT gate [15]. It is a single qubit gate meaning it acts only on one qubit at a time. It is defined as follows in bra-ket notation:

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

Essentially, it inverts the qubit's state. In the diagram, if measurement occurred after the first gate, then this circuit would always return 1.

The second gate is a Hadamard gate [17], which is also a single qubit gate. It is defined as follows:

$$H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$
$$H|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

A Hadamard gate creates an equal superposition. If a qubit is measured directly after a single H gate is applied, it will have an equal probability of returning a 0 or 1. Explained another way, if this circuit ran 100 times, you would expect it to return approximately fifty 0s and fifty 1s. These are just two examples of many quantum gates that can be useful in quantum computation.

1.4.3 Measurement (Output)

The act of measurement signifies the end of quantum computation. In a quantum circuit diagram, measurement is represented by a half circle with an arrow inside a square on the quantum circuit (Figure 1.2).

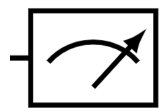


Figure 1.2: The measurement symbol in a quantum circuit diagram.

Though it looks similar to the gate symbols, it is not a quantum gate. Unlike quantum gates, the act of measurement is an irreversible process. During measurement, all qubits' states are collapsed. After measuring the qubits, the circuit outputs a single value (0 or 1) for each qubit. The measurement symbol is often absent from quantum circuit diagrams as it is implied that all qubits are measured at the end of computation.

1.4.4 Error Correction

Theoretically, error correction is not necessary for a quantum circuit. However, in practical systems, error correction is a crucial part of the quantum computation process. It increases the accuracy of our computation and ensures reliable data collection. It remains especially important as quantum computers are expected to be in the NISQ (Noisy Intermediate Scale Quantum) era[20] for the next 5-10 years, making them especially prone to high error rates. For the sake of educational simplicity, we focused on the three prior aspects of the quantum computation process when explaining the quantum circuit. There is remaining work to be done to develop educational material about quantum error correction.

2

Task Analysis

2.1 Introduction

Thus far, we have established the motivation for quantum computing educational visualizations and provided background on key concepts in the field. In this chapter, we will discuss the high-level goals that guided the design of QCVis.

2.2 Domain Goals

QCVis is an experiment in science education. The platform aims to teach students about quantum circuits and quantum computing through scientific visualizations. Intuitively, it is clear that visualizations can aid science education. A visualization provides a visual way to learn about and interpret scientific material that can be more engaging than written text or auditory lectures. Over 20 years of research in the science education field supports these intuitions and proves that visualizations are a necessary part of a science education. Exceptional work conducted by Vavra et. al[30] and Gilbert et. al[11] outlines the main goals of a successful scientific visualization. These goals are as follows:

> **G1**: A successful visualization will promote learning, understanding, analysis and problem solving.
>
> **G2**: A successful visualization will be tested by students who will validate its usefulness.

These goals motivate all aspects of this thesis. They drive the QCVis designs and user studies, detailed in Chapters 3 and 4.

2.3 Domain Driven Tasks

The first goal states that science visualization must teach its students key concepts about the scientific topic to aid in their educational development. Identifying these key concepts in quantum computing took extensive research. We gathered this information using a two-pronged approach. First, we conducted an extensive literature review of the main concepts in the field, as summarized in

Chapter 1. These concepts underlie quantum computing and are important to cover in any quantum computing education. With this knowledge, we quickly realized that it was not within the scope of our project to effectively teach all of these concepts. We needed a way to prioritize which concepts were the most important. For our second approach, we conducted extensive interviews with quantum computing experts and educational professionals. This helped us synthesize all of these concepts into one major takeaway:

> Quantum circuits are the most important concept to teach in a quantum computing educational curriculum.

Quantum circuit diagrams are connected to all of the important quantum computing concepts found in our original analysis. They are an excellent starting point for those new to the field. Furthermore, our experts identified two aspects of a quantum circuit that are essential to understand the concept as a whole. We treated these two aspects as tasks that our educational platform must satisfy in order to create a high quality educational experience. Below are the two tasks that QCVis must complete so that novices can learn, understand, and analyze quantum circuits:

> **T1**: A user must learn how quantum gates change the state probability of a single qubit over time.
>
> **T2**: A user must learn how quantum gates change the state probability of the overall quantum circuit over time.

If our visualization is to achieve the first goal **G1**, then it must successfully teach both of these concepts **T1** and **T2**. The next chapter details the design elements of QCVis and shows how the platform accomplishes the first goal and completes the two tasks. Chapter 4 details the user research findings and explains how QCVis accomplishes the second goal. With these high-level goals and tasks in hand, we can properly analyze the success and shortcomings of QCVis.

3

Design

3.1 Introduction

This chapter describes the implementation and design of the QCVis platform. It highlights the key design elements of the interface and describes how they achieve our first goal and its tasks. The first goal and its connected tasks are as follows:

G1: A successful visualization will promote learning, understanding, analysis and problem solving.

> **T1**: A user must learn how quantum gates change the state probability of a single qubit over time.
>
> **T2**: A user must learn how the quantum gates change the state probability of the overall quantum circuit over time.

The implementation and design of QCVis is one of the two contributions of this thesis.

We present *QCVis*, a novel educational platform that uses two visualization techniques to display state probabilities of both single qubits and quantum circuits over time. Our first visualization technique is a stacked bar chart incorporated into the quantum circuit wire. It displays the probability of a single qubit state over time. Our second visualization technique is a radial bar chart that displays the quantum circuit state probability at a user-specified time in the circuit. The exact time is determined by a time slider which can be moved to any point in the circuit's time. This allows the user to visualize the quantum circuit probability over time.

Over the course of the last year, the QCVis design has undergone many iterations. The first complete version of the platform was called *QuVis* and later became the current QCVis platform. To describe the design of QCVis, we start by describing the implementation and designs of QuVis (Figure 3.1)

The QuVis design was built from February 2020 to October 2020, originally as part of a CS271 course project. This design and preliminary user feedback was published under the 2020 IEEE Quantum Conference. The work for QuVis was done in partnership with Elisa Zhao Hang, Adinawa Adjagbodjou, Robert Krueger, and Johanna Beyer. Chapter 4 details the results of the QuVis user study. Chapter 5 describes the differences between QuVis and QCVis.

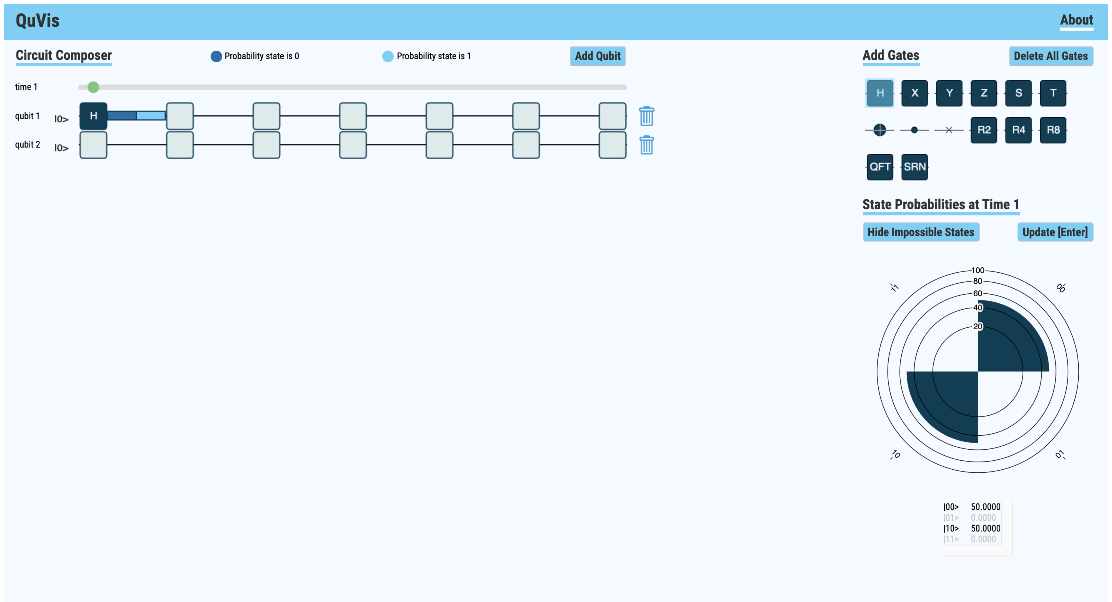


Figure 3.1: The QuVis platform, an early version of QCVis. Circuit Composer displays a H gate with a stackd horizontal barr chart to its right. The State Probabilities panel displays the final state probability of the quantum circuit.

3.2 Implementation

This section describes the implementation of the QCVis platform. While QuVis and QCVis differ in design elements, their implementation architecture and frameworks are the same. QCVis is a web-based platform and accessible on any device with a browser and internet connection. To focus on the educational and visual aspects of the platform, we built our platform on top of QCSimulator[33] an open-source graphic quantum circuit simulator (Figure 3.2).

QCSimulator (Figure 3.2) computes the state probabilities of the quantum circuit using Python linear algebra packages. Its graphical interface is built with CSS and Javascript. Building on top of QCSimulator ensures the platform will have consistent and accurate calculations for the state probabilities. We thank the QCSimulator creators for providing us with this foundational work.

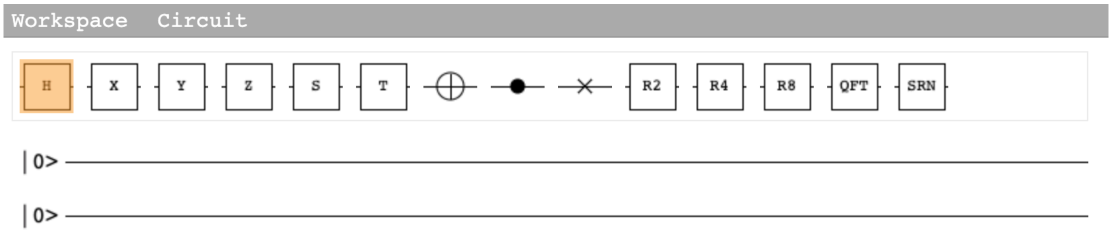


Figure 3.2: An overview of the QCSimulator platform. QCVis was implemented on top of QCSimulator to ensure reliable computation of the state probabilities.

The QCVis platform is implemented with HTML, CSS, Javascript, Python 3 and D3. The HTML, CSS, and Javascript were used to style the platform. The styling included adding colors, text, and interactive button elements. This styling made it easier for users to navigate and interact with the platform. We used Python 3 to calculate the single qubit state probabilities over time. This data eventually fed into the stacked bar charts visualization, discussed more in the Circuit Composer section. Finally, we used D3, a popular Javascript visualization library, to implement the radial bar chart visualization discussed more in the State Probability section.

3.3 QuVis Platform Overview

The following sections outline the key design elements of the QuVis platform. QuVis has two main sections: the Circuit Composer center and the State Probabilities panel. Its two significant contributions are the stacked bar chart and the radial bar chart, located in the Circuit Composer center, and State Probabilities panel, respectively. Below is an example user flow for the QuVis platform:

> A user selects gates from the Add Gates panel and adds them to the quantum circuit in the Circuit Composer center. Immediately, both of our two visualization techniques are displayed. A stacked bar chart appears to the right of the placed gate and displays the state probability of the qubit. Simultaneously, the State Probabilities

panel updates and displays the final state probability of the quantum circuit. A user reads the State Probability panel and understands the output of their quantum circuit.

Beyond the two main contributions, there are additional design considerations behind the QuVis platform discussed at the end of this chapter.

3.3.1 Circuit Composer Center

The goal of the Circuit Composer center is to display the state probability of a single qubit state over time.

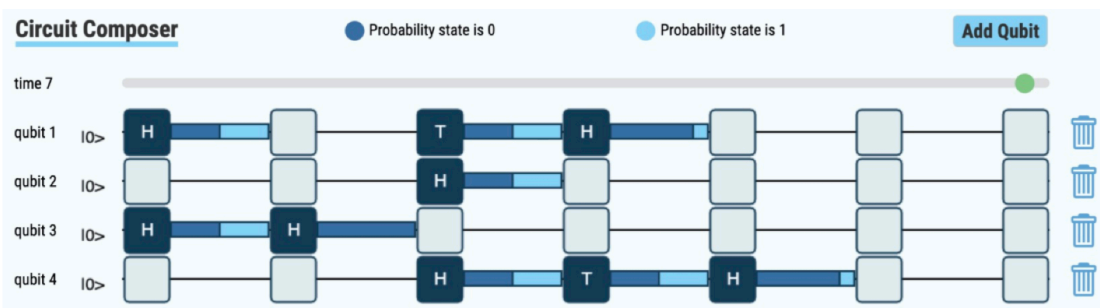


Figure 3.3: The Circuit Composer center with a four qubit circuit, showing how the stacked bar chart changes over time.

In addition, the Circuit Composer center (Figure 3.3) is where users build their quantum circuit. Upon loading the QuVis site, the user is presented with an empty circuit with two qubits. The circuit is displayed as two rows of seven open interconnected boxes. Each of the seven boxes corresponds to a time in the circuit execution. Users can add gates to the circuit via the Add Gates panel.

Within the Circuit Composer center lies one of the two main contributions of QCVis, the stacked bar chart. Between each gate, there are stacked horizontal bar charts that describe each qubit's probability of measuring a 0 or a 1 at that specific timestep. To avoid visual clutter while increasing the density of data, the bar graphs are integrated with the circuit wire. The stacked bar

Figure 3.4: An example of an H gate with a stacked bar chart.

chart is a graphical representation of qubit state superposition, teaching users how qubits can be in both the 0 and 1 states at the same time. As illustrated by the legend at the top of the Circuit Composer center, the dark blue represents the probability that the qubit state is 0, whereas the light blue indicates the probability that the qubit state is 1. In Figure 3.4, the qubit is in an equal superposition of 0 and 1, as shown by the even split of light and dark blue in the bar chart. A user can observe the qubit's state probability over time by noting how the stacked bar chart changes after each gate application. The stacked bar chart visualization is only supported for single qubit gates. Overall, the stacked bar chart visualization helps users understand how a qubit's state probability changes over time, completing the first task **T1**.

Another important component of the Circuit Composer center is the time slider. The time slider is located at the top of the page. It is represented by a vertical bar and a green dot, which a user can drag to input a specific time in the circuit. The time slider teaches users that there are different outputs at different points in time in the quantum circuit. The radial bar chart visualization in the State Probabilities panel depends on the position of this time slider. Users can move the slider to observe how the state probability of the quantum circuit changes over time.

3.3.2 State Probabilities Panel

The goal of the State Probabilities panel is to display the state probability of the quantum circuit over time. This panel is where users see the output of their quantum circuit. Quantum circuit state probability diagrams often suffer from visual clutter. The number of states to display is exponential

(2^n) proportional to the number of qubits. Even a small quantum circuit with 7 qubits will have 128 possible state probabilities to display.

The State Probabilities panel uses a unique radial bar chart to display the quantum circuit's state probability. This radial bar chart (Figure 3.5) is the second contribution of the QuVis platform.

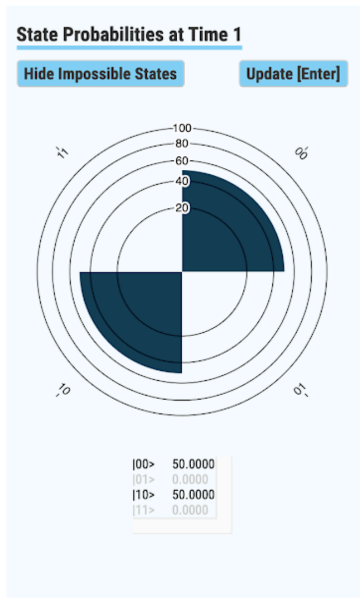


Figure 3.5: State Probability panel showing a superposition of states 10 and 00.

Our radial bar chart attempts to reduce visual clutter in three ways. The first way is embedded in the design of the bar chart. The concentric circles represent different probabilities (eg. 40%, 60%, etc.) and surrounding these circles are all of the possible state probabilities. Each slice of the concentric circle is filled based on the probability of the circuit returning that state. The labels on the radial bar chart correspond to the exact probability. The second way that this visualization reduces visual clutter is through its automated filtering. Users can select "Hide Impossible States" to only show states that have a probability greater than 0. This reduces the total number of states displayed making the visualization easier to read. Lastly, at the bottom of the State Probabilities panel, there is a scrollable list of all the possible states and their exact state probabilities. This ensures that users can access all of the state probabilities, regardless of the size of the circuit. Using all of these visualization tactics to reduce visual clutter, the State Probabilities panel effectively shows users their quantum circuit's state probability at a single timestep.

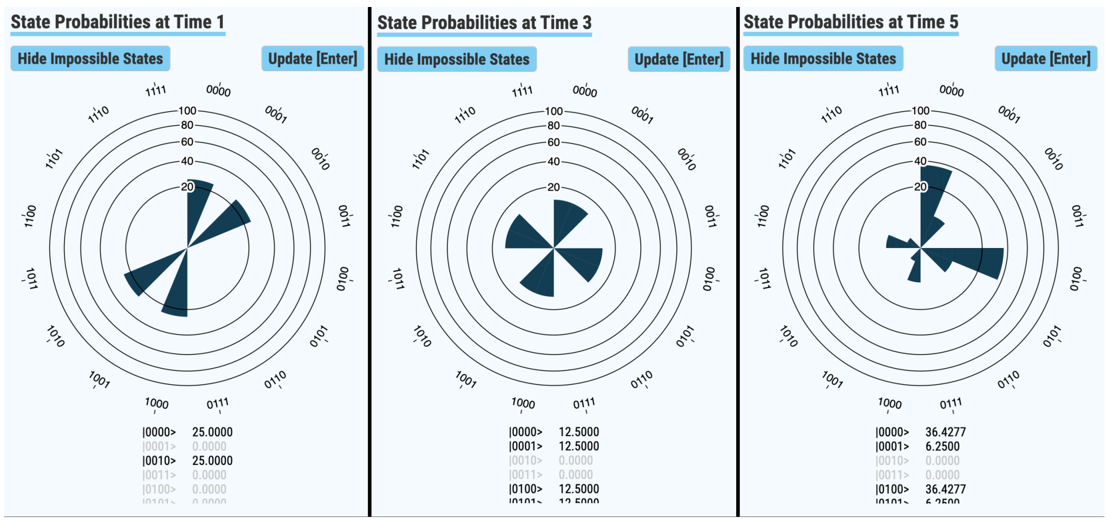


Figure 3.6: State Probability panel changes over time depending on the time slider.

The combination of the State Probabilities panel and time slider ensures users to learn how the state probability changes over time. Updates from the time slider (see Circuit Composer center) are reflected in the radial bar chart. The panel calculates and displays the quantum circuit's state probability up to and inclusive of the current timestep. Figure 3.6 shows how the State Probabilities panel changes over time based on the time slider. Through this discussion, it is clear that QuVis teaches users how the quantum circuit state probability changes over time, completing the second task **T2**.

3.3.3 Additional Design Considerations

While designing QCVis, there were additional features added for functionality and ease of use. One such feature is the Add Gates panel (Figure 3.7). This is where users select gates to add to the

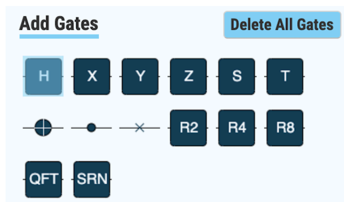


Figure 3.7: The Add Gates panel with all supported quantum gate options.

quantum circuit. Users click to select the available gate options (eg. H, X, etc.) and can place their chosen gate in the grey squares in the Circuit Composer center. This applies the gate to the qubit and adds it to the quantum circuit. Any time a gate is added to the quantum circuit, a stacked bar chart displays to the right of the new gate. (see Circuit Composer Center)

Additionally, we selected a color-blind accessible color palette for QuVis, using Coolors[4], a color scheme analysis tool. This ensures a wide variety of users can successfully navigate the platform. Users can also increase and decrease the size of their quantum circuit in QuVis. To increase the size of the quantum circuit by increasing the number of qubits, users can select the "Add Qubit" button. Similarly, users can select the trash icon to remove a qubit from the circuit. The interface will update to remove a row from bottom of the circuit.

On the backend, QuVis has an event listener which automatically evaluates the circuit whenever a user moves their cursor away from the Circuit Composer. Similarly, if the user moves the time slider to a new value, the circuit is re-evaluated up until that timestep. To manually evaluate a circuit, the user can click the 'Update' button. Each time the circuit is evaluated, the State Probabilities panel instantly updates to reflect the newly calculated distribution. These design considerations for the visualization platform provide real-time feedback and are designed to encourage user exploration of the quantum circuit.

4

User Study

4.1 Introduction

In science education, there are two main goals that test the success of a visualization. The previous chapter describes how QuVis accomplishes the first goal, **G1**, which states a platform must effectively teach the key concepts of the scientific area. This chapter assesses how well QuVis accomplishes **G1** through an examination of **G2**. The second goal of a successful science education visualization is as follows:

> **G2**: A successful visualization will be tested by students who will validate its usefulness.

To achieve this goal, we designed a user study and interviewed six students to validate the usefulness of QuVis. This chapter details the user study procedure, results, and analysis.

4.2 Procedure

This section describes the detailed procedure of our user study. For this study, we interviewed six participants. In visualization research, it is generally accepted that six users can identify the majority of usability problems and successes [16]. All participants were current college students who had at least completed an introductory computer science course. All participants were true novices; they had not encountered quantum computing in any of their coursework and this was their first time learning about the field. Prior to attending the session, users were provided with a brief background on quantum computing and quantum circuits in the form of a blog post [23] and presentation [13].

In our user study, the participants were asked to complete a list of quantum circuit creation and analysis tasks. They completed a similar set of tasks using two quantum circuit creation tools – QCSimulator and QuVis (referred to as Tool A and Tool B in the study). We compared our platform with an existing circuit simulator due to findings from Dow et al.'s work which discovered that people provide more useful feedback when reviewing more than one design [10]. We structured the tasks so that participants would be able to complete them using both tools. Half of our users were randomly assigned to use the QuVis platform first, and the other half used QCSimulator first. Users were encouraged to be descriptive and think out loud as they completed each task.

Each user was asked to perform a set of quantum circuit operations on each platform. First, we asked users to add qubits to the quantum circuit, apply specific gates to qubits at certain timesteps, and evaluate the circuit's output. Then, we asked users to determine the state probability of the quantum circuit at a specific timestep and at the end of the computation. Finally, we asked all participants to create three different quantum circuits that would output a given state probability distribution. An example question we asked is: Create a two qubit circuit, where there is a 100% probability of measuring the state $|11\rangle$. We measured how long it took users to create their final quantum circuit on each platform. This is the target task analyzed in results. After the study, users were asked qualitative questions about both platforms' interfaces and educational quality. In addition, users were asked about the utility of our two main visualizations - the stacked bar chart and the radial bar chart to ensure the features completed **T1** and **T2**. These features were evaluated on a 5-point Likert scale[22]. Lastly, the participants voted on which tool they found more intuitive. The full list of questions can be found in the Appendix.

4.3 Results

Our user research results were highly positive. Overall, users took less time to complete tasks on QuVis than on QCSimulator. Additionally, most users found QuVis more intuitive and would prefer to use it again. Our two main design contributions were also well-received, though users reported that the stacked bar chart was easier to understand than the radial bar chart.

To understand the success of the platform, we compared target task completion times for each participant across both platforms. Regardless of which platform was shown first, five out of the six participants performed the target task faster using QuVis. The average time to complete the task was 40 seconds. The task took 10.21 seconds less when participants used QuVis.

Two out of the three participants who were shown QuVis first took longer to complete similar tasks on QCSimulator. These participants seemed to have difficulty applying the quantum computing intuition they acquired on QuVis to QCSimulator. This suggests that current circuit simulators may be limiting user learnings due to their less intuitive user interface.

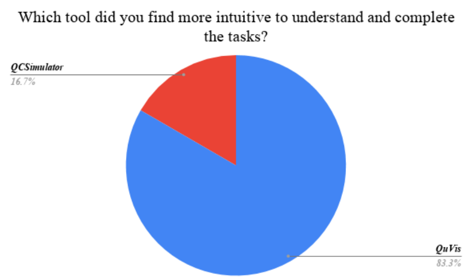


Figure 4.1: User results from the question "Which tool did you find more intuitive to understand and complete the tasks?"

After using QuVis, users had several positive remarks on the platform's usability. Starting with general user feedback, our user study showed that 83% (Figure 4.1) of participants found QuVis more intuitive than QCSimulator. Specifically, users commented positively on the legibility of QuVis, noting the large font, high quality, and frequency of descriptive labels. Users noted the platform had a modern appearance which made it visually appealing and user friendly.

There was positive feedback about QCVis' two main contributions and visualization techniques - the stacked bar chart and radial bar chart. (Figure 4.2)

Users stated that the stacked bar chart was helpful in their understanding of the circuit process. When asked "How helpful was the 'state probabilities' stacked bar chart in Tool B" on a 5-point Likert scale, user response was a mean of 4.00 with a standard deviation of 0.71, demonstrating a positive user experience.

In regards to the radial bar chart, user feedback was mixed. When asked "On a scale of 1-5, how helpful was the 'state probabilities' radial bar chart in Tool B?", user response was a mean of 3.00

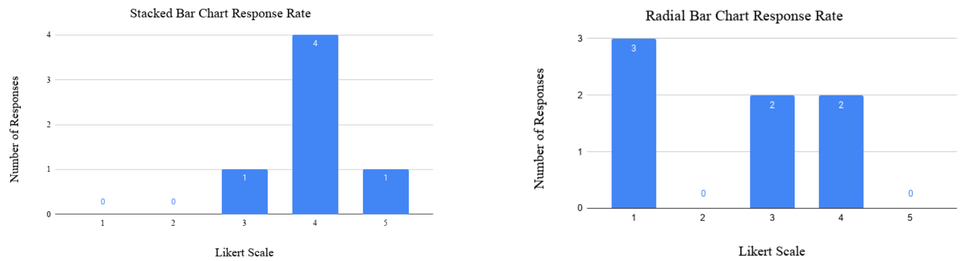


Figure 4.2: User responses histogram of the stacked bar chart and radial bar chart based on the 5-point Likert scale responses.

with a standard deviation of 1.22. Even so, users found it helpful to analyze state probabilities over time, with one user stating that they "like the graphical depiction of state probabilities in Tool B".

There was also insightful user feedback about the time slider. One user commented that "being able to see the state distribution at any timestep was really important". Another user commented how the time slider enhanced the "visual and written depiction of state distribution at any time step". These positive user comments suggest that the time slider was a successful aspect of the QuVis platform. Users also commented favorably about the "Add Qubit" and "Delete" options in the Circuit Composer center.

User research suggests that the dynamic nature of our platform contributed to a more expedient completion of the tasks by helping participants attribute specific actions on the circuit to outcomes they observed. Several users preferred to automatically see changes to the state probabilities in Tool B compared to manually selecting the option to run the circuit in Tool A. One user commented that they "had a good sense of how each action I did (adding a qubit or gate) affected the state probability distribution". This ability to identify how individual components of the interface contribute to the qubit's and quantum circuit's state probabilities over time demonstrates that QuVis achieves **G1**. With these results validated by a user study with novices, QuVis also achieves **G2**. With both goals accomplished, this proves that QuVis is an effective platform for novice quantum computing education.

Even with this demonstrated success, there were still some elements of QuVis that users found confusing. The most common confusion was uncertainty around how to remove gates from the circuit, with one user noting that "I expected the gates to drag and drop". Some participants were confused by the time slider. In particular, one person mentioned that she was "unclear what 'time' referred to". Implementing additional educational documentation around timesteps and the significance of different points in the execution process may help familiarize users. Lastly, we received mixed feedback about the radial bar chart including, "I didn't see the state probability distribution circle... until midway through the exercises". While the Likert scale analysis suggested the radial bar chart was useful for users, users reported some difficulty with the visualization due to its placement on the lower part of the page. Some users did not realize it was there and had difficulty understanding its significance. Moving the panel higher up by swapping its placement with the Add Gates panel may address this concern.

5

QCVis Final Design

5.1 Introduction

Based on user feedback, we have made significant changes to the original QuVis platform. Below is a list of these changes and a presentation of QCVis, the current version of our quantum circuit visualization platform. At the end of this chapter, we offer a few suggestions for future iterations of the platform.

5.2 Design Changes

First, let us discuss the design changes from QuVis to the current version of the platform, QCVis. These changes have been implemented into the current version of QCVis. See Figure 5.1for the new design.

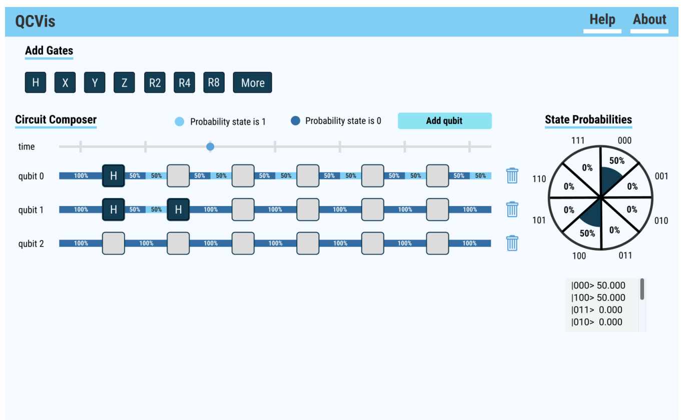


Figure 5.1: The QCVis platform, the contribution of this paper.

The most significant change is in the State Probabilities panel. The State Probabilities panel was the most controversial design element on the QuVis platform. Some users thought it was essential to their understanding, while other users found it confusing. Some of this confusion can be attributed to its placement on the lower half of the screen. To address this issue, we moved the location of the Add Gates panel to ensure the State Probabilities panel can be properly viewed. We also changed the design of the panel to make the state probabilities easier to see by adding lines between neighboring states. This way, even if neighboring states have the same probability, users can still understand how the probability is unique to each state.

Additionally, users also reported some difficulty with the click to select feature in the Add Gates Panel. The panel now has a drag and drop-based implementation. A user can drag a gate onto the Circuit Composer to add it to the quantum circuit. To remove a gate, a user can drag and drop it back onto the Add Gates panel.

The last change is an added educational walkthrough of the platform and its key features. This feature is a new page on the QCVis platform separate from the current home page. It can be accessed by clicking the new "Help" icon on the navigation bar at the top of the screen. This page links to a video walkthrough of the QCVis platform. It explains each design element on the home page through a walkthrough of an example circuit.

An additional user study is required to analyze the efficacy of these additional elements, though preliminary user feedback suggests that educational outcomes have improved.

5.3 Improvements

There are a number of improvements that can be made in future iterations of the QCVis platform. This section outlines a few ideas, though is by no means a comprehensive list of all possible growth areas. One improvement is to the State Probabilities panel. Even with our concerted effort to reduce visual clutter, the State Probabilities panel can still be difficult to read as the number of qubits grows. Currently, the platform offers one filtering option, where it hides all state probabilities equal to 0. However, upon further conversations with experts, there are certain cases where quantum circuits return probabilities that are only slightly greater than 0 (say 0.0002), yet are irrelevant to quantum circuit analysis. In the current version of the platform, these states would still be displayed. An advanced view of the State Probabilities panel, where a user can input the lowest possible probability for displayed states could address this issue. Then, a user could select to only show states greater than 0.0002, thereby eliminating small state probabilities and increasing the readability of the visualization.

Another improvement to the user experience comes from anecdotal user feedback. During user interviews, one user was disappointed that refreshing the page would cause them to lose all of their work on the quantum circuit. Therefore, it may be worthwhile to implement a feature that allows users to save and re-upload their circuits, so they can come back to the platform and expand on previous work. This feature requires additional user validation and expert interviews before it can be implemented.

6
Conclusion

This thesis presents QCVis, a novel educational platform that uses visualization techniques to teach quantum computing concepts. QCVis offers three contributions to the visualization, education, and quantum computing communities. First, it offers a stacked bar chart that aims to visualize and explain how a qubit's state probability changes over time. Second, it offers a radial bar chart that aims to visualize and explain how a quantum circuit's state probability changes over time. Lastly, it offers a user study that aims to quantify the platform's success within its novice target audience. In addition, we established two success goals for all science education visualizations and found that QCVis accomplishes both of these goals.

Prior work has shown that quantum computing is a fast growing technology, with new educational methodologies as a determining factor in its ultimate advancement. This suggests the start of a new field of educational science, that we are on the precipice of a new quest. We hope that QCVis advances this burgeoning field of research and encourages others to join this new quest for knowledge.

A

Appendix

A.1 List of Common Gate Symbols

Below is a list of common gate symbols used in quantum circuit diagrams. Table courtesy of Wikipedia.

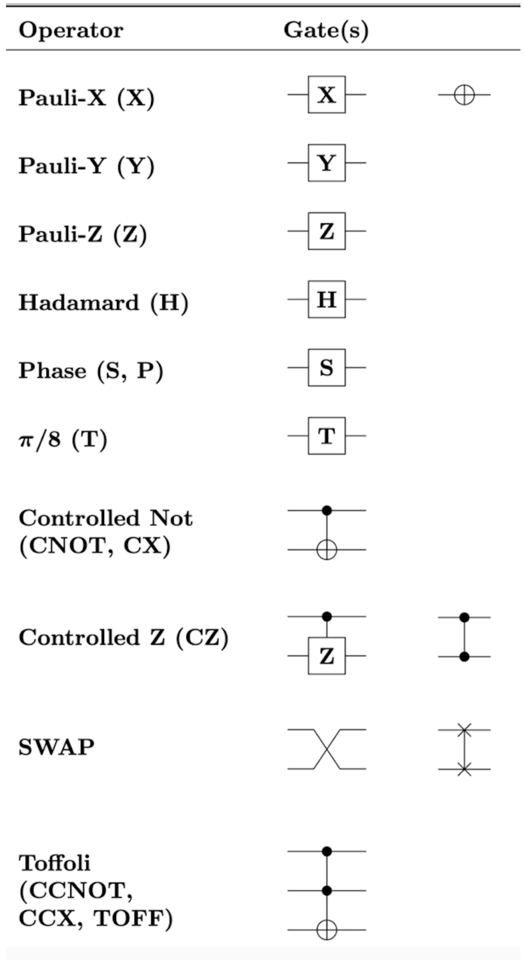


Figure A.1: Common Gate Names and Symbols

A.2 Complete List of User Study Questions

* = This question was used for the time comparison analysis.

A.2.1 Questions for QCSimulator

1. Add three qubits to the circuit.

2. Apply one H gate to each qubit at time = 1.

3. Apply one H gate to each qubit again at time = 2.

4. Run the circuit.

5. What is the state probability distribution at time = 2?

6. Without rerunning the circuit, what is the probability of measuring 1 at time = 1 on qubit 1?

7. Create a two qubit circuit where there is an equal chance of measuring all states ($|00\rangle$, $|10\rangle$, $|01\rangle$, $|11\rangle$). Feel free to use any gates.

8. Create a two qubit circuit where there is a 100% chance of measuring the state $|11\rangle$.

9. Create a three qubit circuit where there is less than a 50% but greater than 0% chance of measuring the state $|101\rangle$.*

A.2.2 Questions for QuVis

1. Add three qubits to the circuit.

2. Apply one H gate to each qubit at time = 1.

3. Apply one H gate to each qubit again at time = 2.

4. Run the circuit.

5. What is the state probability distribution at time = 2?

6. Without rerunning the circuit, what is the probability of measuring 1 at time = 1 on qubit 1?

7. Create a two qubit circuit where there is an equal chance of measuring all states ($|00\rangle$, $|10\rangle$, $|01\rangle$, $|11\rangle$). Feel free to use any gates.

8. Create a two qubit circuit where there is a 100% chance of measuring the state $|10\rangle$.

9. Create a three qubit circuit where there is less than a 50% but greater than 0% chance of measuring the state $|100\rangle$.*

A.2.3 Qualitative Questions

1. What aspects of each user interface were clear to you?

2. Which aspects of each user interface were confusing to you?

3. What aspects of each tool helped you better understand the circuits?

4. On a scale of 1- 5 (Likert scale), how helpful were the stacked bar charts in Tool B?

5. On a scale of 1-5, how helpful was the state probabilities radial bar chart in Tool B?

6. Which tool did you find more intuitive to understand and complete the tasks?

References

[1] Alexander, T., Kanazawa, N., Egger, D. J., Capelluto, L., Wood, C. J., Javadi-Abhari, A., & C McKay, D. (2020). Qiskit pulse: programming quantum computers through the cloud with pulses. *Quantum Science and Technology*, 5(4), 044006.

[2] Arute, F., Arya, K., Babbush, R., & et. al (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574, 505–510.

[3] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., & Weinfurter, H. (1995). Elementary gates for quantum computation. *Phys. Rev. A*, 52, 3457–3467.

[4] Bianchi, F. (2021). Coolors. `https://coolors.co/`.

[5] Bloch, F. (1946). Nuclear induction. *Physical review*, 70(7-8), 460.

[6] Board, N. S. (2020). Science and technology: Public attitudes, knowledge, and interest.

[7] Cory, D. G., Fahmy, A. F., & Havel, T. F. (1997). Ensemble quantum computing by nmr spectroscopy. *Proceedings of the National Academy of Sciences*, 94(5), 1634–1639.

[8] Cross, A. (2018). The ibm q experience and qiskit open-source quantum computing software. *Bulletin of the American Physical Society*, 63.

[9] Dirac, P. A. M. (1939). A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(3), 416–418.

[10] Dow, S., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D., & Klemmer, S. (2011). Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2807–2816).

[11] Gilbert, J., Reiner, M., & Nakleh, M. (2010). *Visualization: theory and practice in science education*. Springer.

[12] Giulini, D. (2008). Electron spin or "classically non-describable two-valuedness". *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 39(3), 557–578.

[13] Hayes, J. (2003). Tutorial:basic concepts in quantum circuits. `http://vlsicad.eecs.umich.edu/BK/Slots/cache/www.eecs.umich.edu/~jhayes/JPH_DACslides_Jun03.pdf`.

[14] Mavroeidis, V., Vishi, K., Zych, M. D., & Jøsang, A. (2018). The impact of quantum computing on present cryptography. *arXiv preprint arXiv:1804.00200*.

[15] Nair, B. S. (2006). *Digital electronics and logic design*. Prentice-Hall of India.

[16] Nielsen, J. & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 249–256).

[17] Nielsen, M. A. & Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge University Press.

[18] Phillips, L. M., Norris, S. P., & Macnab, J. S. (2010). *Visualization in mathematics, reading and science education*. Springer.

[19] Piattini, M. (2020). Training needs in quantum computing.

[20] Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, 2, 79.

[21] Research & Markets (2021). Global quantum computing market with covid-19 impact analysis by offering (systems, services), deployment (on premises, cloud-based), application, technology, end-use industry and region - forecast to 2026.

[22] Robbins, N. B., Heiberger, R. M., et al. (2011). Plotting likert and other rating scales. In *Proceedings of the 2011 Joint Statistical Meeting* (pp. 1058–1066).

[23] Russo, M. (2018). The coming quantum leap in computing.

[24] Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, 56(2), 172–185.

[25] Schumacher, B. (1995). Quantum coding. *Phys. Rev. A*, 51, 2738–2747.

[26] Steane, A. (1998). Quantum computing. *Reports on Progress in Physics*, 61(2), 117–173.

[27] Synder, T. D. (2019). *Digest of education statistics 2017*. NCES Department of Education.

[28] Szafir, D. A. (2018). The good, the bad, and the biased: Five ways visualizations can mislead (and how to fix them). *Interactions*, 25(4), 26––33.

[29] Trimmer, J. D. (1980). The present situation in quantum mechanics: A translation of schrödinger's "cat paradox" paper. *Proceedings of the American Philosophical Society*, 124(5), 323–338.

[30] Vavra, K., Vera Janjic-Watrich, K. L., Phillips, L., Norris, S. P., & Macnab, J. (2011). Visualization in science education. *ASEJ*, 41(1).

[31] Vollmer, H. (2011). *Introduction to circuit complexity: a uniform approach*. Springer.

[32] Winn, W. (1982). Visualization in learning and instruction: a cognitive approach. *ECTJ*, 30, 3–25.

[33] Wybiral, D. (2017). Qc simulator. `https://qcsimulator.github.io/`.

■

Milan Marie Williams